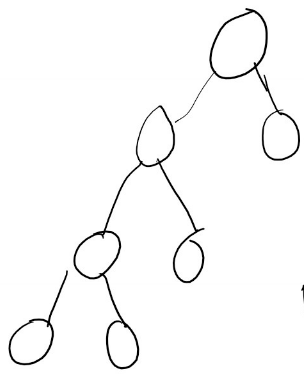
Dylan Munn

CPSC3120: Design Analysis of Algorithms Assignment 2

1. Prove that for a proper binary tree T with n nodes and heigh h, the total number of nodes is at least 2h+1 and at most 2^{h+1} -1

Proper binary tree: Every node other than leaves has two children

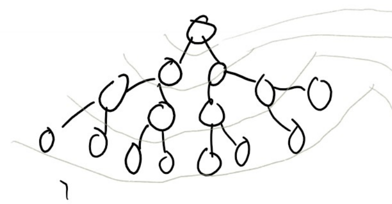
At least 2h+1:



h=0, n=1
h=1, n=2
h=2, n=5
h=3, n=7

As per the definition of proper binary tree, each parent node has to have 2 children. Therefore, each height expansion must mean that there at least 2 nodes added, starting with 1 node (root) at height 0

At most $2^{h+1}-1$:



h=0, n=1
h=1, n=3
h=2, n=7
h=3, n=15

As per the definition of a proper binary tree, you cannot have more than 2 children per parent node. therefore, each height expansion cannot exceed adding 2^h leaves, and internal nodes are at most 2^h-1, as height=0 means 1 root node.

So for each height level, max is $(2^h-1)+2^h \Rightarrow 2^{h+1}-1$

2. Prove that for a proper binary tree T with n nodes and height h, the height is at least log(n+1)-1 and at most (n-1)/2.

Given height of h, we know max nodes is $n = 2h + 1$

- Therefore, for a given node count n, max height is:
  - $n = 2h + 1$
  - $n - 1 = 2h$
  - $\frac{n-1}{2} = h$
  - $h = \frac{n-1}{2}$

Given height of h, we know that minimum nodes is $n = 2^{h+1} - 1$

- Therefore, for a given node count n, minimum height:
  - $n = 2^{h+1} - 1$
  - $n + 1 = 2^{h+1}$
  - $2^{h+1} = n + 1$
  - $h + 1 = \log_2(n+1)$
    - $[y = b^x \Rightarrow x = \log_b(y)]$
  - $h + 1 = \log_2(n+1) - 1$

3. What is the maximum and minimum number of red nodes in a Red-Black tree? Articulate your answer.

Minimum:

- As root is always black, and nodes are never required to be red for tree to be valid, mininum number of red nodes is 0

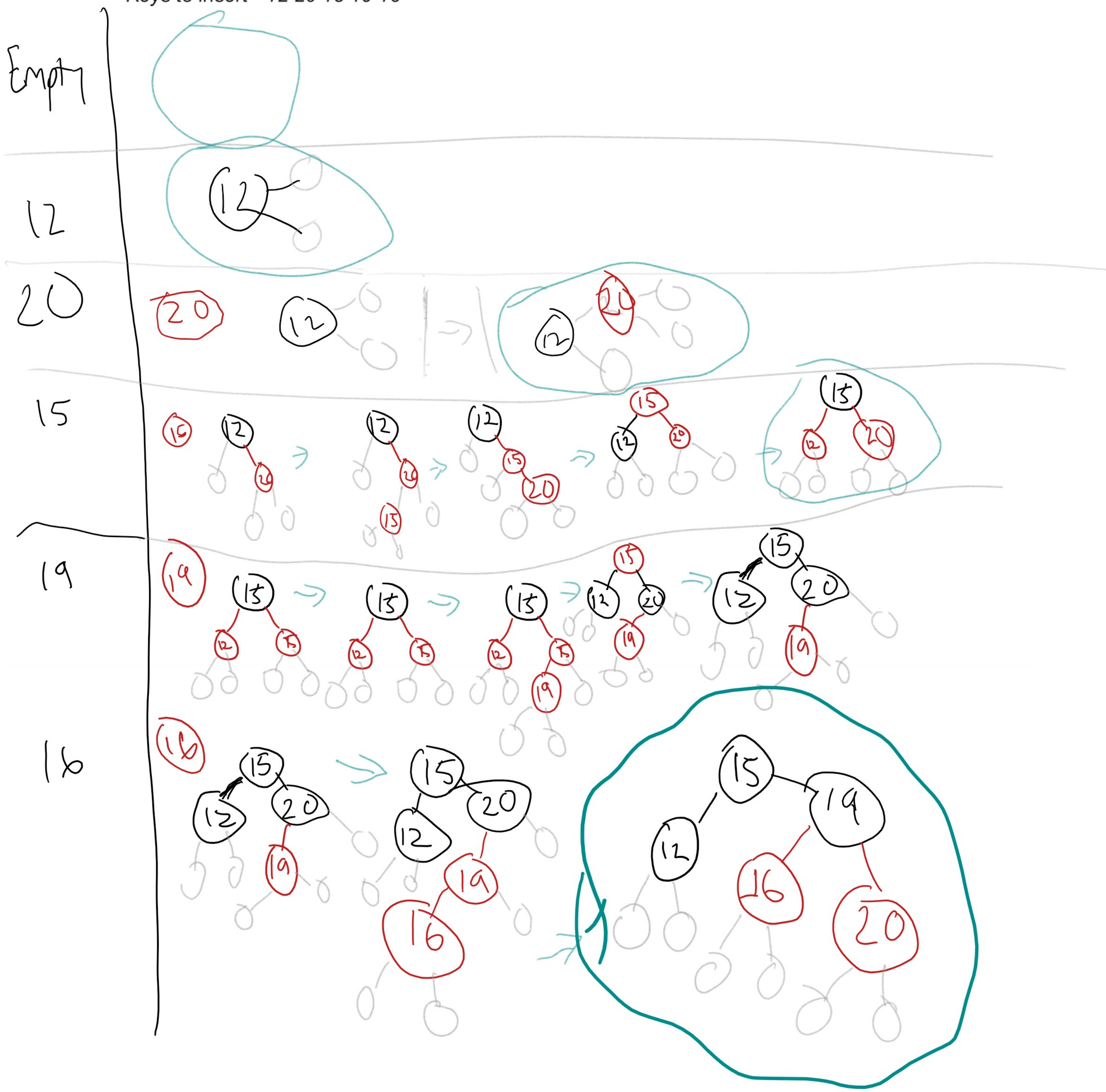
Maximum:

- Given that ...
  - Root always black
  - Can be no more than 2 red children per black node
  - Can be no more than 0 red children per red node
  - For each "depth" of black nodes, there can be 2 red nodes for each parent

- The maximum number of red nodes can be no more than 2x the quantity of black nodes

4. Write the red-black tree that would result from inserting the following list of keys in an initially empty tree T. For each insert, show all the intermediate steps and the transformations applied to the tree.

Keys to insert - 12 20 15 19 16



5. Write the AVL tree that would result from inserting the following list of keys in an initially empty tree T. For each insert, show all the intermediate steps and the transformations applied to the tree.

Keys to insert - 12 20 15 19 16

Empty:

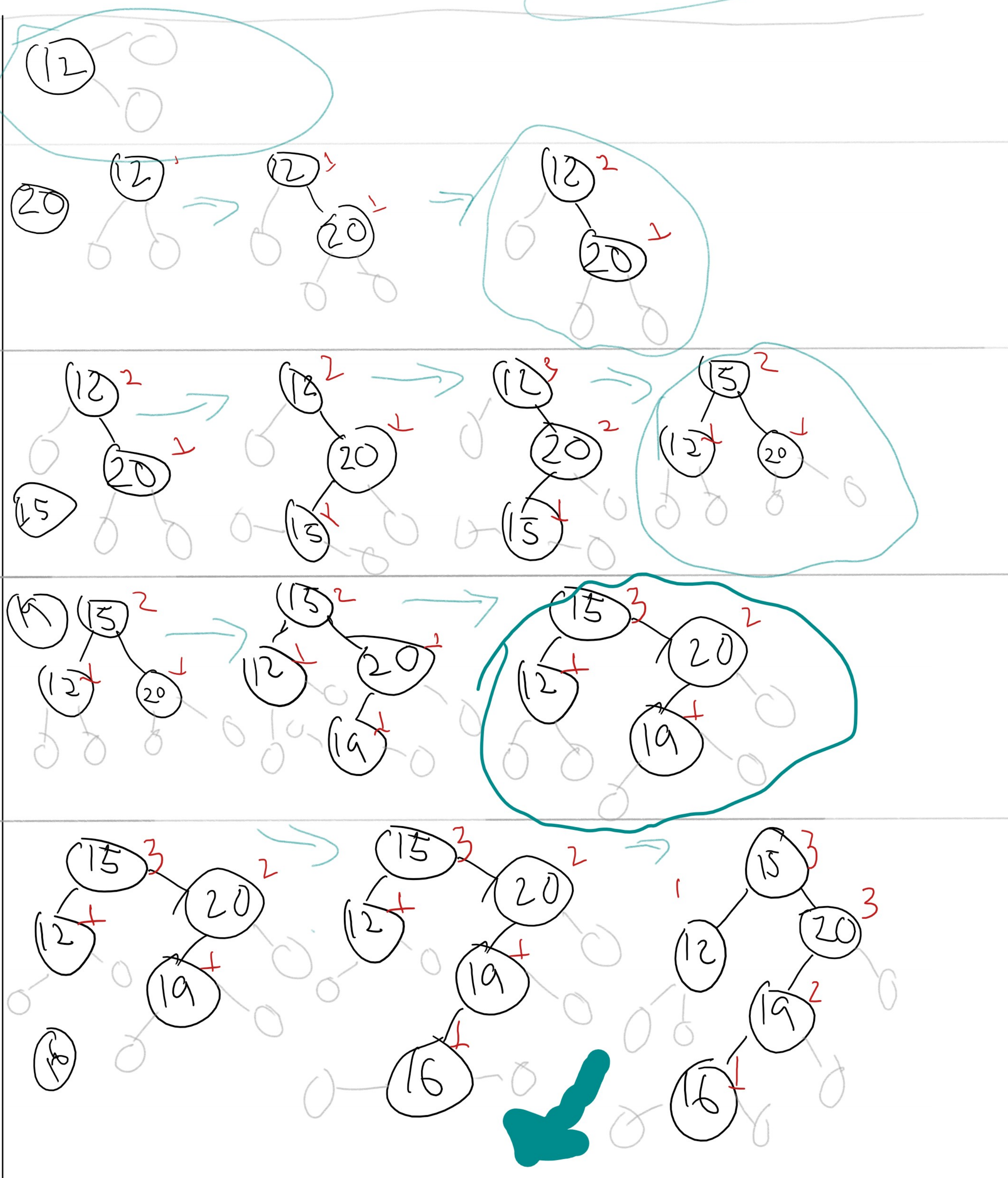
12

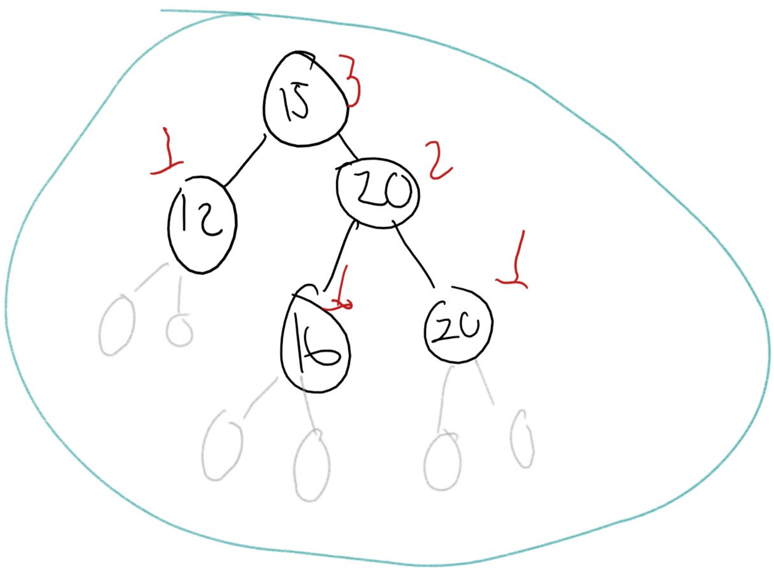
20

15

19

16




15
3
12
1
20
2
16
1
20
1